ELSEVIER

Pattern Recognition Letters 17 (1996) 437–450

Pattern Recognition Letters

New geodesic distance transforms for gray-scale images

Pekka J. Toivanen *

Lappeenranta University of Technology, Department of Information Technology, P.O. Box 20, SF-53851 Lappeenranta, Finland

Received 30 November 1994; revised 15 September 1995

Abstract

In this paper, two new geodesic distance transforms for gray-scale images are presented. The first transform, the Distance Transform on Curved Space (DTOCS), performs the calculation with integer numbers. The second transform, the Weighted Distance Transform on Curved Space (WDTOCS), gives a weighted distance map with real numbers for an arbitrary gray-value image. Both transforms give a distance map in which the distance value of a single point corresponds to the length of the shortest discrete 8-path to the nearest background point. Both differ from the previously presented gray-level distance transforms by not weighting the distance values directly by the gray-values, but by gray-value differences.

Keywords: Distance transforms; Geodesic distance; Grey-level distance transform; Raster scanning; Image processing; Computer vision

1. Introduction

Rutovitz (1968) has proposed an algorithm for obtaining a gray-weighted distance function, in which gray-value is identified with a concept of "height" and gray-weighted distance is defined in such a way that it is less along paths with low gray-value pixels. In his method, he proposes only two iterations, as in the black-and-white case. Rosenfeld (1969) presented a parallel version of this algorithm.

Fall-distance (Rutovitz, 1978; Vossepoel et al., 1979) is another modification of distance, in which the only permitted paths from the reference set are those with falling, i.e. strictly decreasing, gray-values. The set of points reached by such strictly decreasing paths is known as the fall-set of the reference set.

Another generalization, the GRAYMAT (Levi and Montanari, 1970) defines the gray-weighted distance between two points as the smallest sum of gray-levels along any path joining the points. The algorithm is obtained by a suitable generalization of the algorithms that have been used in the black-and-white case, e.g. (Rosenfeld and Pfaltz, 1966; Montanari, 1968). The same algorithm is also presented in (Piper and Granum, 1987) and is used as the first stage of a cost algorithm in (Verbeek and Verwer, 1990)

In (Preteux and Merlet, 1991) two new distance transforms are defined, namely the topographical distance, which is defined using a function called the connection cost, and the differential distance transform, which is defined using the deviation cost function.

[1] Email: pekka.toivanen@lut.fi

PII S0167-8655(96)00010-4

Table 1
The city block kernel

	b	
d	e	f
	h	

Table 2
The 3 × 3 kernel used in this paper

a	b	c
d	e	f
g	h	k

Table 3
The split 3 × 3 kernel used in the forward scan

a	b	c
d	e	

Table 4
The split 3 × 3 kernel used in the backward scan

	e	f
g	h	k

Algorithms for 3D with some generalizations have been described by Mohr and Bajcsy (1983) and Borgefors (1984).

This paper presents two new distance transforms for gray-level images. The first one, called the Distance Transform on Curved Space (DTOCS), performs the distance calculation with integer numbers and gives a distance map, in which the value of every pixel is the length of the shortest path to the nearest background pixel along a discrete 8-path in a square grid. The area in which the transform is calculated may consist of several disjoint regions (Vepsäläinen and Toivanen, 1991; Toivanen, 1993a). The DTOCS and WDTOCS algorithms have been derived from the well-known Rosenfeld–Pfaltz–Laÿ algorithm which calculates the distance transform for binary images and is presented for instance in (Serra, 1988). None of the earlier mentioned transforms calculate the same kind of distance maps as the DTOCS and WDTOCS. All the other gray-weighted skeleton, GRAYMAT etc. algorithms find the minimum path joining two points by the smallest sum of gray-levels or weighting the gray-levels in some manner. The DTOCS does not weight them, it calculates the distance value for each point by calculating the gray-level difference between two adjacent points along the minimal path. The second transform is called the Weighted Distance Transform on Curved Space (WDTOCS). In (Toivanen, 1994) it was called the EDTOCS. It gives a weighted distance map over a gray-value image. Again, every pixel has a distance value corresponding to the length of the shortest discrete 8-path from the pixel to the nearest point in the background.

New image compression algorithms based on DTOCS have been presented in (Toivanen, 1992, 1993a,b).

2. Definition of the DTOCS

In the distance map produced by the Distance Transform on Curved Space (DTOCS), every pixel has a distance value which corresponds to the distance of that pixel from the nearest background pixel along a path according to the following definitions.

Definition 1. Let $X \subset \mathbb{Z}^2$. Let $B \subset \mathbb{Z}^2$ be the structuring element. Let the external boundary of X be denoted by ∂X and be defined by $\partial X = (X \oplus B) \setminus X$. $\partial X \subset X^C$. See (Giardina and Dougherty, 1988).

In the definitions of the DTOCS and the WDTOCS below we will use the following notation. $x \in X$ and $y \in \partial X$. Let $\Psi_X(x,y)$ be the set of digital 8-paths in $(X \cup \partial X)$ linking x and y. Let $\gamma \in \Psi_X(x,y)$ and let γ have n pixels. Let $a_i \in \gamma$ and $a_{i+1} \in \gamma$ be two adjacent pixels in the path γ. Let $\mathcal{G}_X(a_i)$ denote the gray-value of the pixel a_i. The Distance Transform on Curved Space (DTOCS) is defined as follows.

Definition 2. Let the distance between a_i and a_{i+1} be $d_X(a_i, a_{i+1}) = |\mathcal{G}_X(a_i) - \mathcal{G}_X(a_{i+1})| + 1$, $i = 1, 2, \ldots, n-1$. The length of the path γ is defined by $\Lambda(\gamma) = \sum_{i=1}^{n-1} d_X(a_i, a_{i+1})$. The DTOCS distance image is defined by

$$\mathcal{F}_X(x) = \min(\Lambda(\gamma), \gamma \in \Psi_X(x,y)), \quad y \in \partial X,\ \Psi_X(x,y) \neq \emptyset, \tag{1}$$

$$\mathcal{F}_{\partial X}(y) = 0. \tag{2}$$

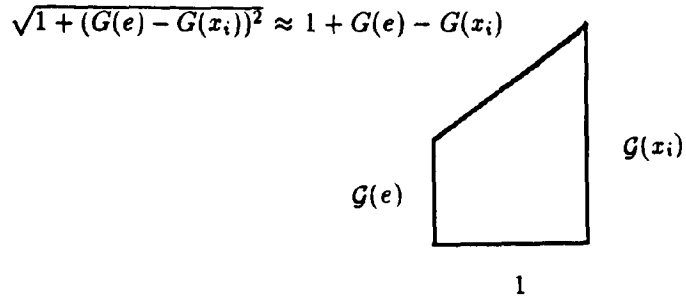

Fig. 1. The height displacement of DTOCS for all neighbors of a pixel e, i.e. $x_i \in N_8(e)$ in a rectangular grid.

3. The kernels

In the city block kernel (see Table 1), the diagonal corner points are omitted. It is assumed that the distance from them to point e is infinite. Due to the lack of points, the results obtained by the city block kernel are poorer than those obtained by the full kernel from the earlier stated definition of the DTOCS point of view.

The 3×3 kernel used in this paper is depicted in Table 2. In the 2-phase algorithm, the kernel in Table 3 is used in the forward scan and the one in Table 4 in backward scan.

Borgefors (1986) analyzed the behaviour of the 5×5 kernel for binary images, and therefore those results are not applicable here. The question of applying bigger than 3×3 kernels is left to be analyzed in the future.

4. The DTOCS algorithm

Let $X \subset \mathbb{Z}^2$. Let $x \in X$ and $y \in X$ be two points in 2-dimensional discrete space. A sequential two-pass algorithm to calculate the Distance Transform on Curved Space is presented in (Vepsäläinen, 1991) and (Toivanen, 1992). The following algorithm requires two images: the original gray-level image $\mathcal{G}(x)$ and a binary image $\mathcal{F}(x)$ which determines the region(s) in which the transform is performed. In $\mathcal{F}(x)$, X is initialized to the maximal representative number of the memory and X^C to 0 similarly as with the the Rosenfeld–Pfaltz–Laÿ algorithm. See (Serra, 1988). $\mathcal{F}^*(x)$ means an already calculated point. $\mathcal{F}^*(e)$ denotes the new distance value of the point e in the image $\mathcal{F}$. It should be noted that the region X in which the following transform is performed may consist of several disjoint regions. This applies also to the background area X^C. The following two-pass algorithm is also easily adapted to the hexagonal grid in the same way as the Rosenfeld–Pfaltz–Laÿ algorithm. The rectangular grid displacement 1 is replaced by the integer approximation of the height displacement

$$\sqrt{1+(G(e)-G(x_i))^2} \approx 1+G(e)-G(x_i)$$

for all neighbors x_i of a pixel e ($x_i \in N_8(e)$) as seen in Fig. 1. $N_8(e)$ denotes the 8 neighbors of pixel e in a rectangular grid. The constant 1 has to be added to the gray-value difference in the same way as 1 is added to the pixel value in the Rosenfeld–Pfaltz–Laÿ algorithm. Otherwise the DTOCS algorithm will not give a distance map at all.

In Eqs. (4) and (6), the parameter α governs the amount in which the curvature of the gray-level image $\mathcal{G}$ is taken into account. If $\alpha = 1$, the algorithms will give the DTOCS distance map according to the Definitions 1 and 2. If $\alpha = 0$, then Eqs. (3)–(6) will reduce to the Rosenfeld–Pfaltz–Laÿ algorithm (Serra, 1988). If $\alpha = 0.5$, every pixel value in the distance map will hold a value which is obtained by dividing by two the value which is obtained when $\alpha = 1$. In other words, the distance values are proportional to α. This is stated and proved in Theorem 1. In order to work the following algorithm requires that $maxint + 1 = 0$ in the same way as the Rosenfeld–Pfaltz–Laÿ algorithm does.

Table 5
Pixel values of an arbitrary 5 × 5 neighborhood in image $\mathcal{G}(x,y)$. The same neighborhood is also in image $\mathcal{F}(x,y)$

$\mathcal{G}(i-2,j-2)$	$\mathcal{G}(i-2,j-1)$	$\mathcal{G}(i-2,j)$	$\mathcal{G}(i-2,j+1)$	$\mathcal{G}(i-2,j+2)$
$\mathcal{G}(i-1,j-2)$	$\mathcal{G}(i-1,j-1)$	$\mathcal{G}(i-1,j)$	$\mathcal{G}(i-1,j+1)$	$\mathcal{G}(i-1,j+2)$
$\mathcal{G}(i,j-2)$	$\mathcal{G}(i,j-1)$	$\mathcal{G}(i,j)$	$\mathcal{G}(i,j+1)$	$\mathcal{G}(i,j+2)$
$\mathcal{G}(i+1,j-2)$	$\mathcal{G}(i+1,j-1)$	$\mathcal{G}(i+1,j)$	$\mathcal{G}(i+1,j+1)$	$\mathcal{G}(i+1,j+2)$
$\mathcal{G}(i+2,j-2)$	$\mathcal{G}(i+2,j-1)$	$\mathcal{G}(i+2,j)$	$\mathcal{G}(i+2,j+1)$	$\mathcal{G}(i+2,j+2)$

First iteration

The first iteration round proceeds in the "direct video order" (from top to bottom, and from left to right) calculating the new point $\mathcal{F}^*(e)$. The points marked with asterix * hold already once calculated distance values while the point $\mathcal{F}(e)$ has the initial value, which is the maximal representative integer number. With the kernel of Table 3 the first iteration proceeds as follows.

$$\mathcal{F}^*(e) = \min[\mathcal{F}(e), \min(1 + da + \mathcal{F}^*(a), 1 + db + \mathcal{F}^*(b), 1 + dc + \mathcal{F}^*(c), 1 + dd + \mathcal{F}^*(d))], \quad (3)$$

where

$$da = \alpha|\mathcal{G}(e) - \mathcal{G}(a)|, \quad db = \alpha|\mathcal{G}(e) - \mathcal{G}(b)|, \quad dc = \alpha|\mathcal{G}(e) - \mathcal{G}(c)|, \quad dd = \alpha|\mathcal{G}(e) - \mathcal{G}(d)|. \quad (4)$$

The new $\mathcal{F}(e)$ ($\mathcal{F}^*(e)$) is calculated from already calculated points $\mathcal{F}(a), \mathcal{F}(b), \mathcal{F}(c), \mathcal{F}(d)$ and the corresponding subtractions of $\mathcal{G}(a), \mathcal{G}(b), \ldots$ from $\mathcal{G}(e)$.

Second iteration

The second iteration round proceeds in the "inverse video order" (from bottom to up, and from right to left) calculating the new point $\mathcal{F}^*(e)$. The points marked with asterix * hold already calculated distance values while the point $\mathcal{F}(e)$ has a value obtained when applying Eq. (2). With the kernel of Table 4 the second iteration proceeds as follows.

$$\mathcal{F}^*(e) = \min[\mathcal{F}(e), \min(1 + df + \mathcal{F}^*(f), 1 + dg + \mathcal{F}^*(g), 1 + dh + \mathcal{F}^*(h), 1 + dk + \mathcal{F}^*(k))], \quad (5)$$

where

$$df = \alpha|\mathcal{G}(e) - \mathcal{G}(f)|, \quad dg = \alpha|\mathcal{G}(e) - \mathcal{G}(g)|, \quad dh = \alpha|\mathcal{G}(e) - \mathcal{G}(h)|, \quad dk = \alpha|\mathcal{G}(e) - \mathcal{G}(k)|. \quad (6)$$

Theorem 1. *Let $\delta_X(x,y)$ be the distance value between points $x \in X$ and $y \in X^C$ given by the DTOCS algorithm if $\alpha = 1$. If $0.0 < \alpha < 1.0$, the distance $d_X(x,y)$ given by DTOCS between all points x and y is proportional to α.*

$$\forall x \in X,\ \forall y \in X^C: \quad d_X(x,y) = \alpha\,\delta_X(x,y). \quad (7)$$

Proof. Let the images $\mathcal{F}(x,y)$ and $\mathcal{G}(x,y)$ be two-dimensional buffers. Let the 3 × 3 kernel (Table 2) be at an arbitrary place in the image $\mathcal{F}(x,y)$, which is the distance image, and in the $\mathcal{G}(x,y)$ image, which is the original gray-value image. Let the center point of the kernel, e, be at $\mathcal{F}(i,j)$ in the distance image and at $\mathcal{G}(i,j)$ in the gray-level image. The first pixel a of the kernel lies at $\mathcal{G}(i-1,j-1)$ and $\mathcal{F}(i-1,j-1)$. The last pixel k lies at $\mathcal{F}(i+1,j+1)$ and $\mathcal{G}(i+1,j+1)$. See Table 5.

Consider Eq. (5):

$$\mathcal{F}^*(e) = \min[\mathcal{F}(e), \min(1 + df + \mathcal{F}^*(f), 1 + dg + \mathcal{F}^*(g), 1 + dh + \mathcal{F}^*(h), 1 + dk + \mathcal{F}^*(k))].$$

The first term on the right-hand side of =, $\mathcal{F}(e)$, is obtained at the first iteration step and is given by Eq. (3). It is inserted into Eq. (5):

$$\mathcal{F}^*(e) = \min[\min[\mathcal{F}(e), \min[1 + \alpha|\mathcal{G}(e) - \mathcal{G}(a)| + \mathcal{F}(a), \ldots]],$$
$$\min(1 + \alpha|\mathcal{G}(e) - \mathcal{G}(f)| + \mathcal{F}(f), \ldots)].$$

According to the pixel coordinates in Table 5, $\mathcal{F}(e) = \mathcal{F}(i,j)$, $\mathcal{G}(e) = \mathcal{G}(i,j)$ etc. Elaborating $\mathcal{F}(f)$ gives

$$\mathcal{F}(f) = \min[\mathcal{F}(i,j+1), \min(1 + \alpha|\mathcal{G}(i,j+1) - \mathcal{G}(i-1,j)| + \mathcal{F}(i-1,j), \ldots).$$

Inserting this into the equation gives

$$\mathcal{F}^*(e) = \min[\min[\mathcal{F}(i,j), \min[1 + \alpha|\mathcal{G}(i,j) - \mathcal{G}(i-1,j-1)| + \mathcal{F}(i-1,j-1), \ldots]],$$
$$\min[(1 + \alpha|\mathcal{G}(i,j) - \mathcal{G}(i,j+1)|$$
$$+ \min[\mathcal{F}(i,j+1), \min(1 + \alpha|\mathcal{G}(i,j+1) - \mathcal{G}(i-1,j)| + \mathcal{F}(i-1,j), \ldots)]].$$

Since $\mathcal{F}(i,j) = \mathcal{F}(i-1,j-1) = \mathcal{F}(i,j+1) = \mathcal{F}(i-1,j) = maxint$, it follows that

$$\mathcal{F}^*(e) = \min[\min[1 + \alpha|\mathcal{G}(i,j) - \mathcal{G}(i-1,j-1)| + maxint, \ldots],$$
$$\min[(1 + \alpha|\mathcal{G}(i,j) - \mathcal{G}(i,j+1)| + \min[1 + \alpha|\mathcal{G}(i,j+1) - \mathcal{G}(i-1,j)| + maxint, \ldots]]].$$

Since $maxint + 1 = 0$, it follows that

$$\mathcal{F}^*(e) = \min[\alpha|\mathcal{G}(i,j) - \mathcal{G}(i-1,j-1)|, \ldots),$$
$$\min[(\alpha|\mathcal{G}(i,j) - \mathcal{G}(i,j+1)| + \min(\alpha|\mathcal{G}(i,j+1) - \mathcal{G}(i-1,j)|, \ldots)]],$$
$$\mathcal{F}^*(e) = \min[(\alpha|\mathcal{G}(i,j) - \mathcal{G}(i-1,j-1)|, \ldots),$$
$$\min[(\alpha|\mathcal{G}(i,j) - \mathcal{G}(i,j+1)| + \alpha\min(|\mathcal{G}(i,j+1) - \mathcal{G}(i-1,j)|, \ldots)]],$$
$$\mathcal{F}^*(e) = \min[(\alpha|\mathcal{G}(i,j) - \mathcal{G}(i-1,j-1)|, \ldots),$$
$$\min[\alpha(|\mathcal{G}(i,j) - \mathcal{G}(i,j+1)| + |\mathcal{G}(i,j+1) - \mathcal{G}(i-1,j)|), \ldots]],$$
$$\mathcal{F}^*(e) = \alpha\min[(|\mathcal{G}(i,j) - \mathcal{G}(i-1,j-1)|, \ldots),$$
$$(|\mathcal{G}(i,j) - \mathcal{G}(i,j+1)| + |\mathcal{G}(i,j+1) - \mathcal{G}(i-1,j)|, \ldots)],$$
$$\mathcal{F}^*(e) = \alpha\min[\ldots]. \quad \square$$

Corollary. *Eqs.* (3) *and* (5) *can be written in the following way:*

$$\textit{first iteration:} \quad \mathcal{F}^*(e) = \alpha\min[\mathcal{F}(e), \min(1 + da + \mathcal{F}^*(a), \ldots)], \tag{8}$$
$$\textit{second iteration:} \quad \mathcal{F}^*(e) = \alpha\min[\mathcal{F}(e), \min(1 + df + \mathcal{F}^*(f), \ldots)], \tag{9}$$

where

$$da = |(\mathcal{G}(e) - \mathcal{G}(a))|, \quad \ldots, \quad df = |(\mathcal{G}(e) - \mathcal{G}(f))|, \quad \ldots.$$

Proof. The proof is the same as the proof of Theorem 1. $\square$

5. Definition of the WDTOCS

Let the discrete 8-path denote a possible discrete path linking two points according to the 8 neighbors of every pixel in the square grid. The Weighted Distance Transform on Curved Space (WDTOCS) between two points is defined as the minimum of all possible paths linking those points. Along this path, each subdistance

Fig. 2. The height displacement of EDTOCS for all 4 horizontal and vertical neighbors for a pixel e, i.e. $x_i \in N_4(e)$.

Fig. 3. The height displacement of EDTOCS for all 4 diagonal neighbors of a pixel e.

betweeen two points is Euclidean, but the whole distance is not. Using Definition 1 and the same notation as with DTOCS, the WDTOCS is defined as follows.

Definition 3. Let the distance between a_i and a_{i+1} be

$$d_X(a_i, a_{i+1}) = \sqrt{(\mathcal{G}(a_i) - \mathcal{G}(a_{i+1}))^2 + 1}, \quad i = 1, 2, \ldots, n-1, \quad \text{if } a_{i+1} \in N_4(a_i);$$
$$d_X(a_i, a_{i+1}) = \sqrt{(\mathcal{G}(a_i) - \mathcal{G}(a_{i+1}))^2 + 2}, \quad i = 1, 2, \ldots, n-1, \quad \text{if } a_{i+1} \in (N_8(a_i) \setminus N_4(a_i)).$$

See Figs. 2 and 3. The length of the path γ is $\Lambda(\gamma) = \sum_{i=1}^{n-1} d_X(a_i, a_{i+1})$. The WDTOCS distance image is defined by

$$\mathcal{F}_X(x) = \min(\Lambda(\gamma), \gamma \in \Psi_X(x, y)), \quad y \in \partial X, \quad \Psi_X(x, y) \neq \emptyset, \tag{10}$$
$$\mathcal{F}_{\partial X}(y) = 0. \tag{11}$$

6. The WDTOCS algorithm

The WDTOCS algorithm was first introduced in (Toivanen, 1994) under the name EDTOCS. It requires only two passes over the image with a chosen kernel. In order to implement the WDTOCS algorithm, two surface models are needed: the original gray-level image, and another, which determines the region or regions in which the transform is calculated. The transform is performed on this image. The part of the surface where the distance function is calculated, X, is initialized to maximal representative number of the memory and its complement X^C to 0. It should be noted that the region X in which the following transform is performed may consist of several disjoint regions. The algorithm, which applies the WDTOCS, proceeds as follows. Let $\mathcal{G}(x)$ denote the original gray-level image and let $\mathcal{F}(x)$ denote the binary image which determines the region(s) in which the transform is calculated. $\mathcal{F}^*(x)$ means an already calculated point. $\mathcal{F}^*(e)$ denotes the new distance value of the point e in the image $\mathcal{F}$. Let $N_4(e)$ denote the 4 horizontal and vertical neighbors of a pixel e similarly as in the city block kernel. Fig. 2 shows the Euclidean distance between pixel e and its 4 neighbors $N_4(e)$. $G(e)$ denotes the gray-value of the center point in the 3×3 kernel and $G(x_i)$ denotes the gray-values of the pixels $x_i \in N_4(e)$. Fig. 3 shows the same for all the 4 diagonal neighbors of a pixel e.

First iteration

The first iteration round proceeds in the "direct video order" (from top to bottom, and from left to right) calculating the new point $\mathcal{F}^*(e)$. The points marked with asterix * hold already once calculated distance values

while the point $\mathcal{F}(e)$ has the initial value, which is the maximal representative integer number. Using the kernel of Table 3 the iteration proceeds as follows.

$$\mathcal{F}^*(e) = \min[\mathcal{F}(e), \min(da + \mathcal{F}^*(a), db + \mathcal{F}^*(b), dc + \mathcal{F}^*(c), dd + \mathcal{F}^*(d))], \tag{12}$$

where

$$\begin{aligned} da &= \alpha\sqrt{(\mathcal{G}(e) - \mathcal{G}(a))^2 + \beta}, & db &= \alpha\sqrt{(\mathcal{G}(e) - \mathcal{G}(b))^2 + \delta}, \\ dc &= \alpha\sqrt{(\mathcal{G}(e) - \mathcal{G}(c))^2 + \beta}, & dd &= \alpha\sqrt{(\mathcal{G}(e) - \mathcal{G}(d))^2 + \delta}. \end{aligned} \tag{13}$$

In (Toivanen, 1994) $\beta = 2$ and $\delta = 1$ corresponding to the WDTOCS definition presented in Definitions 1 and 3. Borgefors (1986) presented optimal propagating weights for a binary case 3×3 mask. These weights are $\beta = 1.36930$ and $\delta = 0.95509$.

Second iteration

The second iteration round proceeds in the "inverse video order" (from bottom to up, and from right to left) calculating the new point $\mathcal{F}^*(e)$. The points marked with asterix * hold already once calculated distance values while the point $\mathcal{F}(e)$ has a value obtained when applying Eqs. (12) and (13). Using the kernel of Table 4 the second iteration proceeds as follows.

$$\mathcal{F}^*(e) = \min[\mathcal{F}(e), \min(df + \mathcal{F}^*(f), dg + \mathcal{F}^*(g), dh + \mathcal{F}^*(h), dk + \mathcal{F}^*(k))], \tag{14}$$

where

$$\begin{aligned} df &= \alpha\sqrt{(\mathcal{G}(e) - \mathcal{G}(f))^2 + \delta}, & dg &= \alpha\sqrt{(\mathcal{G}(e) - \mathcal{G}(g))^2 + \beta}, \\ dh &= \alpha\sqrt{(\mathcal{G}(e) - \mathcal{G}(h))^2 + \delta}, & dk &= \alpha\sqrt{(\mathcal{G}(e) - \mathcal{G}(k))^2 + \beta}. \end{aligned} \tag{15}$$

Again, in (Toivanen, 1994) $\beta = 2$ and $\delta = 1$ corresponding to the WDTOCS definition presented in Definitions 1 and 3. Borgefors (1986) presented optimal propagating weights for a binary case 3×3 mask. These weights are $\beta = 1.36930$ and $\delta = 0.95509$.

Theorem 2. *Eqs.* (12) *and* (14) *are equal to the following equations:*

$$\textit{first iteration:} \quad \mathcal{F}^*(e) = \alpha \min[\mathcal{F}(e), \min(da + \mathcal{F}^*(a), \ldots)], \tag{16}$$

$$\textit{second iteration:} \quad \mathcal{F}^*(e) = \alpha \min[\mathcal{F}(e), \min(df + \mathcal{F}^*(f), \ldots)], \tag{17}$$

where

$$da = \sqrt{(\mathcal{G}(e) - \mathcal{G}(a))^2 + \beta}, \quad \ldots, \quad df = \sqrt{(\mathcal{G}(e) - \mathcal{G}(f))^2 + \delta}, \ldots .$$

Proof. The proof has been presented in (Toivanen, 1994). □

7. Performance of the DTOCS and the WDTOCS

Definition 4. Let one iteration step denote applying Eq. (3), (5), (12), or (14). Let one iteration round denote applying two iteration steps, i.e. Eqs. (3) and (5) for DTOCS, or (12) and (14) for WDTOCS.

Definition 5. The correct distance map of DTOCS denotes a distance map which is constructed according to the DTOCS definitions (Definitions 1 and 2), and the correct distance map of WDTOCS denotes a distance map which is constructed according to the WDTOCS definitions (Definitions 1 and 3).

Table 6
The upper part of the original image

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	2	2	2	2	0	0	0	0	0	0	2	2	2	2	2	0
0	2	3	3	3	2	0	0	0	0	0	0	2	3	3	3	2	0
0	2	3	4	3	2	1	0	0	0	0	0	2	3	4	3	2	0
0	2	3	4	3	2	1	0	0	0	1	1	2	3	4	3	2	0
0	2	3	4	3	2	1	1	0	1	1	1	2	3	4	3	2	0
0	2	3	4	3	2	1	1	1	1	1	1	2	3	4	3	2	0

Table 7
The distance map obtained by GRAYMAT presented in (Levi, 1970)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	10	10	10	10	10	0	0	0	0	0	0	10	10	10	10	10	0
0	10	35	35	35	10	0	0	0	0	0	0	10	35	35	35	10	0
0	10	35	70	45	20	5	0	0	0	0	0	10	35	70	35	10	0
0	10	35	70	45	20	5	0	0	0	5	5	20	45	70	35	10	0
0	10	35	70	55	30	15	5	0	5	15	15	30	55	70	35	10	0
0	10	35	70	65	40	25	15	5	15	25	25	40	65	70	35	10	0

Table 8
The distance map obtained by the Gray-Weighted Distance Transform presented in (Rutovitz, 1968) and later used in (Piper, 1987) and (Verwer, 1990)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	2	2	2	2	0	0	0	0	0	0	2	2	2	2	2	0
0	2	5	5	5	2	0	0	0	0	0	0	2	5	5	5	2	0
0	2	5	9	5	2	1	0	0	0	0	0	2	5	9	5	2	0
0	2	5	9	5	3	1	0	0	0	1	1	2	5	9	5	2	0
0	2	5	9	6	3	1	1	0	1	1	2	3	5	9	5	2	0
0	2	5	9	6	3	2	1	1	1	2	2	4	6	9	5	2	0

Table 9
The distance image obtained by DTOCS with split 3 × 3 kernel

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	3	3	3	3	3	0	0	0	0	0	0	3	3	3	3	3	0
0	3	5	5	5	3	0	0	0	0	0	0	3	5	5	5	3	0
0	3	5	7	5	3	2	0	0	0	0	0	3	5	7	5	3	0
0	3	5	7	5	4	2	0	0	0	2	2	3	5	7	5	3	0
0	3	5	7	6	4	2	2	0	2	2	3	4	5	7	5	3	0
0	3	5	7	6	4	3	2	2	2	3	3	5	6	7	5	3	0

The performance of DTOCS and WDTOCS is tested using real-world images. Table 6 shows part of the original image taken from (Levi, 1970). The corresponding distance map produced by the Gray-Weighted Medial Axis Transform (GRAYMAT) presented in (Levi, 1970) is shown in Table 7. The distance map that DTOCS produces on the image of Table 6 is different. See Table 9. This difference is clearly noticeable when comparing Tables 7 and 9. Table 8 shows the distance map produced by the Gray-Weighted Distance Function presented in (Rutovitz, 1968). The same algorithm is also presented in (Piper, 1987) and it is used as the first stage of a cost algorithm in (Verwer, 1990). Also this algorithm produces a different distance map compared to DTOCS. The distance values are lower than DTOCS values at points which hold low gray-values, and are higher than DTOCS values at points which hold greater gray-values than DTOCS. In other words, the method presented in (Rutovitz, 1968; Piper, 1987; Verwer, 1990) calculates a distance map in which the distance values are weighted by gray-values. The DTOCS does not weight the distance values directly by gray-values. For instance, consider the last row of the original image. The pixel in lower left corner (Table 6) has a gray-value 0. The next point on the right hand side is 2. The DTOCS distance between these pixels is then $2 + 1 = 3$ (Table 9: column 2, last row), which is the distance value of the point. The GRAYMAT gives 10 (Table 7) and the Gray-Weighted Distance Transform gives 2 (Table 8). Furthermore, consider the 4th column of the tables. The DTOCS distance is 7 (see column 4 in the last row of Table 6). The GRAYMAT gives 70 and the Gray-Weighted Distance Transform gives 9, which is clearly bigger than the DTOCS value and a result from weighting the distance values by gray-levels.

The performance of the DTOCS and WDTOCS are somewhat dependable on the nature of the gray-level image on which they are performed. According to the tests made, the DTOCS and WDTOCS almost always give a distance map according to their definitions for small neighborhoods after the first iteration round, i.e., after applying Eqs. (3) and (5) once for DTOCS, and Eqs. (12) and (14) once for WDTOCS. If they are applied to

Table 10
The original image. Pixels marked -59-, -64-, etc. denote background pixels, i.e. pixels belonging to X^C

107	59	59	69	64	66	66	68	78
115	70	70	76	69	75	68	63	81
111	64	64	73	67	68	67	69	65
113	65	65	75	74	71	67	71	67
114	66	66	74	70	70	63	60	66
112	59	-59-	-64-	66	67	-66-	62	62
116	60	60	58	60	67	-64-	59	55
110	65	65	70	57	61	68	64	63

Table 11
DTOCS distance map after the first iteration round. Erroneous pixels are marked with letter x and pixels belonging to X^C with –

52	26	25	14	18	15	15	13	24
57	14	14	16	13	19	12	15	25
52	7	7	12	12	11	10	12	13
52	5	5	12	10	7	9	13	10
53	4	3	11	6	5	4	6	8
x	1	–	–	3	2	–	3	x
x	2	2	2	5	2	–	6	11
53	8	8	13	4	4	4	1	3

Table 12
The distance image after the 2nd iteration round

52	26	25	14	18	15	15	13	24
57	14	14	16	13	19	12	15	25
52	7	7	12	12	11	10	12	13
52	5	5	12	10	7	9	13	10
53	4	3	11	6	5	4	6	8
51	1	–	–	3	2	–	3	4
56	2	2	2	5	2	–	6	11
53	8	8	13	4	4	4	1	3

Table 13
The original spiral image. The calculation is performed relative to point -0-, which is the only point belonging to X^C

0	0	0	0	0	0	0
0	11	11	11	11	11	0
0	0	0	0	0	11	0
0	11	11	11	0	11	0
0	11	-0-	0	0	11	0
0	11	11	11	11	11	0
0	0	0	0	0	0	0

Table 14
DTOCS distance map after the first iteration round. The point belonging to X^C is marked with –

7	7	19	18	17	16	15
6	16	15	15	15	14	14
6	5	4	3	3	13	14
18	12	12	12	2	12	13
17	12	–	1	2	11	12
16	12	12	12	13	12	12
15	15	14	13	12	12	12

Table 15
The DTOCS distance map after 2 iteration rounds

7	7	8	9	10	11	12
6	16	15	15	15	15	12
6	5	4	3	3	14	13
6	12	12	12	2	14	14
7	12	–	1	2	14	15
8	12	12	12	13	14	14
9	9	10	11	12	13	14

bigger areas, more than one iteration round is needed. Also, if the image is complicated, for instance, the area X is composed of several disjoint regions with spirals or other difficult patterns, more than one iteration round is needed. As a result, a distance map is generated. This distance map can again be inserted to the algorithm replacing the original binary image $\mathcal{F}(x)$. Again, a distance map is obtained. If the map generated by the first iteration round was correct according to the definition presented for DTOCS in Definitions 1 and 2, and for WDTOCS in Definitions 1 and 3, this map equals the first map and does not change during successive iteration rounds. Otherwise, some or all the errors in the first map will be corrected during the second iteration round. The number of iteration rounds needed depends on the image, but according to the tests made, the DTOCS and WDTOCS algorithms almost always converge to the right distance maps according to their definitions. All tests have been run on a Convex 3420 (ConvexOS 10.1) UNIX workstation with no optimizing or vectorizing options. In (Piper, 1987) the need for several iteration rounds for other existing sequential local distance transforms is dicussed.

Table 10 shows the original gray-value image, which has been arbitrarily taken from the "Leena" image. The distance map generated by the DTOCS after the first iteration phase, i.e. applying Eqs. (3) and (5) once, is shown in Table 11. Points that hold erroneous distance values are marked with letter x and the reference points, i.e. points belonging to X^C, with -. Table 12 depicts the same distance map after the second iteration phase. Now the map is correct, and further iterations, if applied to the distance image, would not change the map.

Table 13 shows a spiral image, which can be considered rather difficult for the two-pass DTOCS algorithm, since the distance values may become trapped inside the spiral resulting in false distance values. The reference point, i.e. the only point belonging to X^C, is marked with -0-. Table 14 shows the distance image obtained

Table 16
The original image. The calculation is done relative to points marked with -31-, -36-, etc., i.e. points belonging to X^C

53	56	63	60	38	43	51
56	43	48	47	40	38	74
49	38	34	33	37	60	77
36	-31-	-36-	40	53	-65-	66
34	48	43	60	72	-70-	70
45	55	53	76	68	83	80
55	70	59	68	71	79	55

Table 17
WDTOCS distance image after the 1st iteration round. Erroneous points are marked with x. X^C points are marked with –

17.648	20.587	27.582	24.632	10.586	13.775	18.141
20.505	7.549	12.549	x	10.458	9.172	10.952
13.495	2.450	2.236	3.317	7.440	5.099	12.083
x	–	–	4.123	12.042	–	1.414
3.3166	12.083	7.071	10.574	2.236	–	1.000
14.362	17.162	17.121	x	2.450	13.038	10.100
x	13.823	12.919	3.864	5.612	11.832	21.860

Table 18
The WDTOCS distance map after 2 iteration rounds. Now the map is correct with respect to the definition of the WDTOCS. The X^C points are marked with – (they hold a distance value 0.0)

17.648	20.587	27.582	24.632	10.586	13.775	18.141
20.505	7.549	12.549	13.963	10.458	9.172	10.952
13.495	2.450	2.236	3.317	7.440	5.099	12.083
4.899	–	–	4.123	12.042	–	1.414
3.3166	12.083	7.071	10.574	2.236	–	1.000
14.362	17.162	17.121	6.479	2.450	13.038	10.100
18.576	13.823	12.919	3.864	5.612	11.832	21.860

Table 19
The WDTOCS distance map obtained using in Eqs. (13) and (15) the values $\beta = 1.36930$ and $\delta = 0.95509$

17.608	20.549	27.548	24.595	10.475	13.728	18.040
20.476	7.514	12.517	13.900	10.430	9.106	10.868
13.465	2.424	2.216	3.298	7.410	5.090	12.078
4.848	–	–	4.112	12.038	–	1.383
3.298	12.078	7.065	10.540	2.216	–	0.955
14.339	17.072	17.110	6.444	2.424	13.035	10.093
18.441	13.712	12.844	3.793	5.572	11.789	21.805

after the first iteration round. Now the points near the reference point are correct according to the definition of the DTOCS. After applying the DTOCS algorithm (Eqs. (3), (4), (5) and (6)) twice, i.e. after the second iteration round, the errors disappear. Table 15 depicts the final distance map, which is correct according to the definition of the DTOCS.

Table 16 shows the original image. It has been taken randomly from the well-known "Leena" image. The calculation is performed relative to four points belonging to X^C and marked with -31-, -36-, etc. These four points form two disjoint regions. The first region consists of points -31- and -36-. The second region consists of points -65- and -70-. Table 17 depicts the WDTOCS distance map after the first iteration round, i.e., after applying Eqs. (12), (13), (14) and (15) once. Points which hold erroneous values are marked with letter x. X^C points are marked with -. Table 18 shows the distance map after the second iteration round. Now the map is correct, i.e. every point $p \in X$ has a distance value corresponding to the shortest path from the point to the nearest point in the background X^C according to the WDTOCS definition. Table 19 shows the WDTOCS distance map obtained when in Eqs. (13) and (15) $\beta = 1.36930$ and $\delta = 0.95509$.

When using the weights $\beta = 1.36930$ and $\delta = 0.95509$ in Eqs. (13) and (15), the WDTOCS algorithm gives a better approximation to the Euclidean distance map than with the weights $\beta = 1$ and $\delta = 1$. Still it overestimates the Euclidean distance. The weights 1.36930 and 0.95509 are used as propagating displacements on the image plane. 1.36930 is used for the diagonal directions and 0.95509 for the rectangular directions, as in (Borgefors, 1986). The height displacement remains the gray-value difference.

Table 20 shows the difference between the distances generated by the WDTOCS and the distances obtained by the DTOCS for both horizontal and vertical, and diagonal points of the 8 neighboring pixels of any pixel x in the image $\mathcal{G}(x)$. The DTOCS approximates the WDTOCS always to the following greater integer number. The first column, G, denotes the difference between gray-values of x and its 8 neighboring pixels.

Fig. 4 shows how the DTOCS converges to the correct distance map for different image sizes. The well-known "Leena" girl image of different sizes was used as the original gray-level image. The 3×3 kernel of Tables 3 and 4 was used. For instance, with an image of size 32×32 pixels, the number of erroneous pixels on different iteration rounds was 213, 193, 99, 34, 1 and 0. Fig. 5 illustrates the convergence properties of the WDTOCS algorithm with normal 3×3 kernel of Tables 3 and 4. For a 32×32 image the number of erroneous pixels on different iteration rounds was 413, 207, 101, 27, 1 and 0.

Table 20
Comparison between DTOCS and EDTOCS for all pixels $q \in N_8(p)$ in a square grid

G	WDTOCS distances horizontal and vertical points	WDTOCS distances diagonal points	DTOCS distances all 8 points
0	1	1.41	1
1	1.41	1.73	2
2	2.23	2.45	3
3	3.16	3.31	4
4	4.12	4.24	5
5	5.10	5.19	6
6	6.08	6.16	7
7	7.07	7.14	8
8	8.06	8.12	9
9	9.05	9.11	10

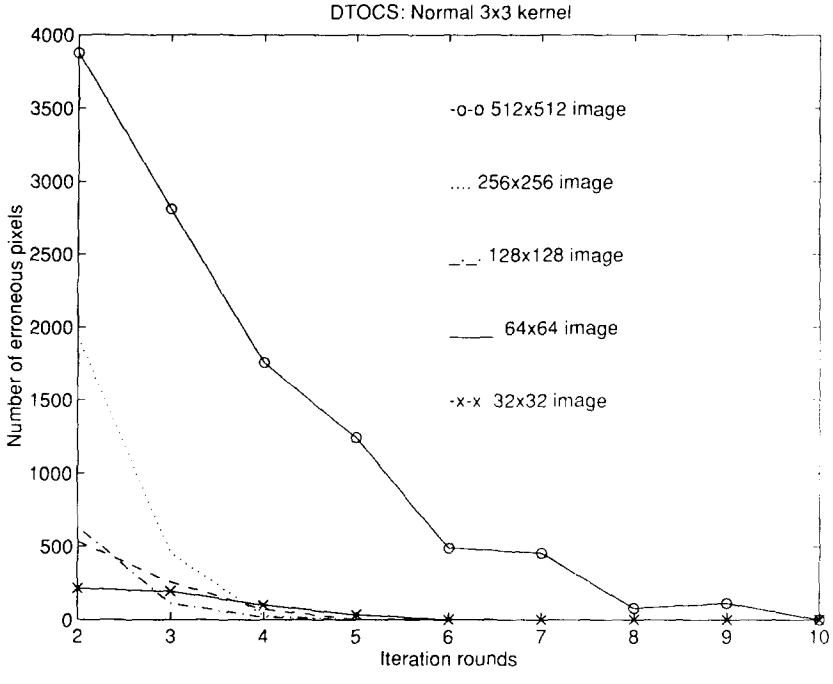

Fig. 4. Convergence of the DTOCS for different image sizes. Normal 3 × 3 kernel.

The DTOCS and WDTOCS algorithms have a time complexity of approximately $O(n^2)$ for an $n \times n$ image, i.e. the same complexity as the GRAYMAT (Levi, 1970) and the gray-weighted distance function proposed in (Rutovitz, 1968). Fig. 6 shows the CPU time vs. image size for DTOCS and WDTOCS. Note that for instance the image size 100 means a 100 × 100 image.

Features of these algorithms are summarized as follows:

1. The DTOCS gives a gray-level weighted version of the chessboard distance. The weights are not constant, but gray-value differences of the original image. In the DTOCS distance map, every point has a distance value corresponding to the length of the shortest path to the nearest background point. The minimal paths linking two points are discrete 8-paths.
2. The WDTOCS is a weighted distance transform with real numbers which propagates local Euclidean distance weights inside a kernel. The weights are not constant, but gray-value differences of the original image. These differences are computed in a different way compared to DTOCS. The obtained whole distance between two

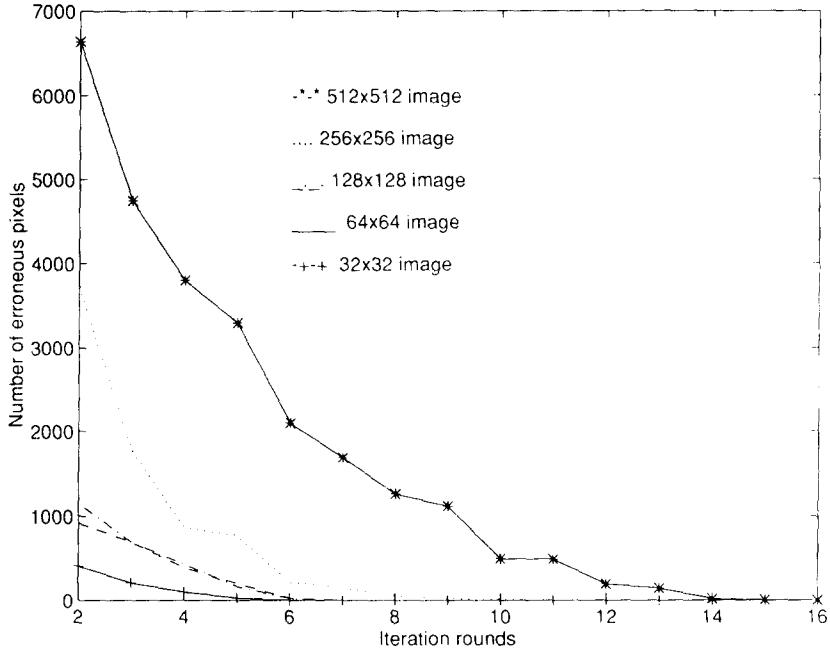

Fig. 5. Convergence of the WDTOCS for different image sizes. Normal 3 × 3 kernel.

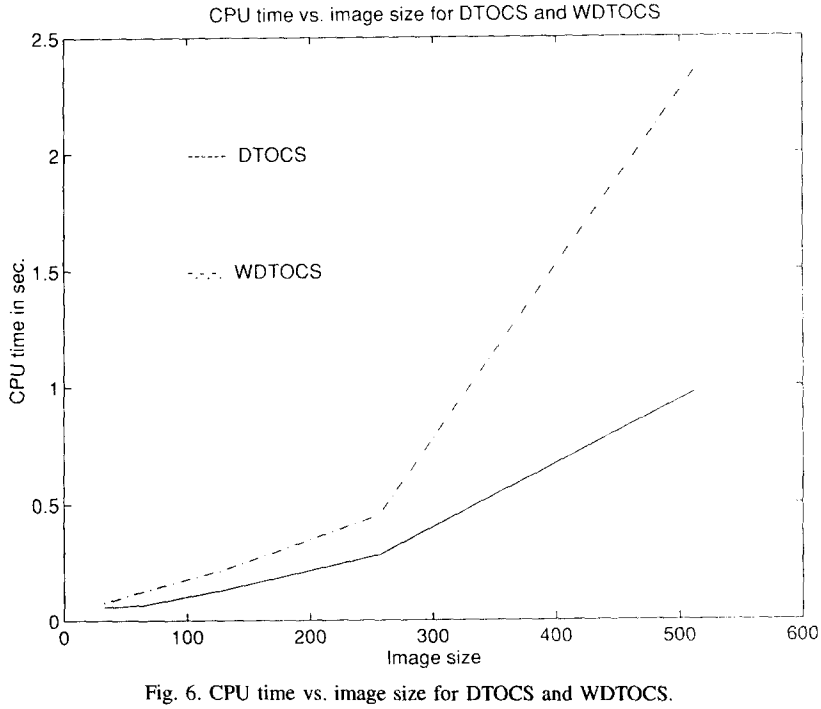

Fig. 6. CPU time vs. image size for DTOCS and WDTOCS.

points is not Euclidean, but each subdistance along the minimal discrete 8-paths is. The WDTOCS gives a better approximation to the Euclidean distance if the optimal binary case weights are used as part of the propagating weights.

3. One iteration round requires only two passes over the image.
4. It is easily adaptable to other grids. For example, the hexagonal grid is quite straightforward.
5. Only two image buffers are needed: the original gray-value image and the binary image which defines the region(s) of calculation.
6. The parameter α governs the amount in which the curvature is taken into account.

8. Conclusion

This paper presents two new distance transforms for gray-level images. The first one, called the Distance Transform on Curved Space (DTOCS), performs the distance calculation with integer numbers and gives a weighted distance map for an arbitrary gray-level image, in which the value of every pixel is the length of the shortest path to the nearest background pixel. The area in which the transform is calculated may consist of several disjoint regions. The second transform is called the Weighted Distance Transform on Curved Space (WDTOCS). It gives a weighted distance map over a gray-level image. Again, the value of every pixel in the distance map denotes the length of the shortest discrete 8-path path to the nearest background pixel. Along the 8-path, every subdistance between two neighboring pixels is Euclidean, but the whole distance is not. The WDTOCS gives a better approximation to the Euclidean distance if the optimal binary case weights are used as part of the propagating weights inside the kernel.

It is shown that both the DTOCS and WDTOCS converge to the correct distance map with respect to their definitions in a few iteration rounds, i.e. when applying the two-pass algorithm 3–10 times. The number of iteration rounds depends on the nature of the image and on the size of the image in which the transform is performed. The best results are obtained by the ordinary 3×3 kernel which is split for forward and backward scans. The city block kernel gives poorer results.

None of the earlier developed transforms (Rutovitz, 1968; Levi, 1970; Piper, 1987; Verwer, 1990) calculate the same kind of distance maps as the DTOCS and WDTOCS do. All the other gray-weighted distance function, GRAYMAT etc. algorithms find the minimum path joining two points by the smallest sum of gray-levels or weighting the distance values directly by the gray-levels in some manner. The DTOCS does not weight them that way. The DTOCS gives a gray-level weighted version of the chessboard distance map. The weights are not constant, but gray-value differences of the original image. The difference between the DTOCS and WDTOCS is that the WDTOCS calculates these gray-level differences in a different way.

Besides the image compression applications presented in (Toivanen, 1992, 1993a, b) the DTOCS and WDTOCS can be used in calculating minimal distances in digitized surfaces and in minimal path-finding problems.

Acknowledgements

The author wishes to thank M.Sc. Leena Tamminen, M.Sc. Jouni Ikonen and Mr. Janne Marin for help.

References

Borgefors, G. (1984). Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing* 27, 321–345.

Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing* 34, 344–371.

Giardina, C.R. and E. R. Dougherty (1988). *Morphological Methods in Image and Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ.

Levi, G. and U. Montanari (1970). A grey-weighted skeleton. *Inform. Control* 17, 62–91.

Mohr, R. and R. Bajcsy (1983). Packing volumes by spheres. *IEEE Trans. Pattern Anal. Mach. Intell.* 5 111–116.

Montanari, U. (1968). A method for obtaining skeletons using a quasi-euclidean distance. *J. Assoc. Comput. Mach.* 15, 600–624.

Piper, J. and E. Granum (1987). Computing distance transformations in convex and non-convex domains. *Pattern Recognition* 20 (6), 599–615.

Preteux, F. and N. Merlet (1991). New concepts in mathematical morphology: the topographical distance functions. In: P.D. Gader and E.R. Dougherty, Eds., *Proc. SPIE* 1568, 66–77.

Rosenfeld, A. and J. L. Pfaltz (1966). Sequential operations in digital picture processing. *J. Ass. Comput. Mach.* 13, 471–494.

Rosenfeld, A. and J. Pfaltz (1968). Distance functions on digital pictures. *Pattern Recognition* 1 (1), 33–61.

Rosenfeld, A. (1969). *Picture Processing by Computer*. Academic Press, New York, 142.

Rutovitz, D. (1968). Data sructures for operations on digital images. In: G. C. Cheng, R. S. Ledley, D. K. Pollock and A. Rosenfeld, Eds., *Pictorial Pattern Recognition*. Thompson Book, WA, 105–133.

Rutovitz, D. (1978). Expanding picture components to natural density boundaries by propagation methods, the notions of fall-set and fall-distance. *Proc. 4th Internat. Joint Conf. on Pattern Recognition*, Kyoto, Japan, November 7–10, 1978, 657–664.

Serra, J. (1988). *Image Analysis and Mathematical Morphology, Vol. 2: Theoretical Advances*. Academic Press, New York.

Toivanen, P.J. (1992). Fast image compression using distance function on curved space. In: *Proc. SPIE Visual Communications and Image Processing '92* 1818, 877–884.

Toivanen, P.J. (1993a). Image compression using DTOCS and neighborhood masks along threshold boundaries. In: *Proc. IEEE Winter Workshop on Nonlinear Digital Signal Processing '93*, Tampere, Finland, 2.2-6.1–2.2-6.6.

Toivanen, P.J. (1993b). Image compression by selecting control points using distance function on curved space. *Pattern Recognition Lett.* 14, 475–482.

Toivanen, P.J. (1994). The Euclidean Distance Transform on Curved Space (EDTOCS) with application to image compression. In: *Proc. 7th European Signal Processing Conf. (EUSIPCO-94)*, Edinburgh, Scotland, UK, 983–986.

Toivanen, P.J. (1995). Convergence properties of the Distance Transform on Curved Space (DTOCS). In: *Proc. 1995 Finnish Signal Processing Symposium (FINSIG'95)*, Helsinki, Finland, 75–79.

Vepsäläinen, A.M. and P.J. Toivanen (1991). Two new image compression methods utilizing mathematical morphology. In: *Proc. SPIE Visual Communications and Image Processing '91* 1606, 282–293.

Verbeek, P.W. and B.J.H. Verwer (1990). Shading from shape, the eikonal equation solved by gray-weighted distance transform. *Pattern Recognition Lett.* 11, 681–690.

Vossepoel, A.M., A.W.M. Smeulders and K. Van der Broek (1979). DIODA: delineation and feature extraction of microscopical objects. *Comp. Prog. Biomedicine* 10, 231–244.